AC自动机

2019年8月20日 21:22

1. 洛谷的自动机

```
struct AC{
    int ch[MN][mn];     //字典树
    int val[MN]; //字符串结尾标号
    int f[MN];   //fail指针，指向失配后与失配字母前一字母相同的结点或根
    int tot;            //最晚插入的结点号
    inline void init(){ tot=0, ms(ch[0],0), val[0] =0; }//初始化首结点
    void ins(char*s,int v){   //把s插进ch，val值为v
        int p=0;     //当前结点指针
        int len=strlen(s);
        for_(i,0,len){
            int id=s[i]-'a';
            if(!ch[p][id])      //需申请新结点
                ch[p][id] =++tot,
                ms(ch[tot],0),
                val[tot] =0;
            p= ch[p][id];
        }
        val[p] =v;
    }
    void build(){//构造f数组
        queue<int>q;
        f[0] =0;
        int p;
        for_(i,0,mn)
            if(p= ch[0][i])     //有指向字母i的结点
                f[p] =0, q.push(p);
        while(q.size()){
            int t= q.front();  q.pop();
            for_(i,0,mn)
                if(p= ch[t][i])     //t存在指向i的结点
                    q.push(p), f[p] = ch[ f[t] ][i];
                else ch[t][i] = ch[ f[t] ][i];
        }
    }
    int find(char*s){   //返回找到了几个不重单词，去掉注释就输出找到几次
        int p=0;
        int len= strlen(s);
        int ans=0;
        for_(i,0,len){      //遍历句中每个字母
            p= ch[p][s[i] -'a'];
            for(int t=p; t/*&&~val[t]*/; t= f[t])
                ans+= val[t]? 1: 0  /*,val[t]=-1*/;
        }
    }
};
```

2. 并不是很会用的、程序设计之门的自动机

```
struct AC{
    int ch[MN][mn];     //字典树
    int val[MN]; //字符串结尾标号
```

```
    int f[MN];   //fail指针，指向失配后与失配字母前一字母相同的结点或根
    int lst[MN]; //lst[i]=j表示j是i的后缀
    int sz;                    //相当于tot，指向最晚插入的位置
    inline void init(){ sz=1, ms(ch[0],0), val[0] =0; } //初始化首结点
    void insert(char*s,int v){//把s插进ch，val值为v
        int p=0;     //当前结点指针
        int len=strlen(s);
        for_(i,0,len){
            int id=s[i]-'a';
            if(!ch[p][id])      //需申请新结点
                ch[p][id] =sz,
                ms(ch[sz],0),
                val[sz++] =0;
            p= ch[p][id];
        }
        val[p] =v;
    }
    void getfail(){     //构造f和lst数组
        queue<int>q;
        lst[0] =f[0] =0;
        int p;
        for_(i,0,mn)
            if(p= ch[0][i])
                f[p] =lst[p] =0, q.push(p);
        while(q.size()){
            int t= q.front();  q.pop();
            for_(i,0,mn){
                if(p= ch[t][i]){
                    q.push(p);
                    int ft= f[t];
                    while(ft && !ch[ft][i]) ft= f[ft];
                    f[ft] = ch[ft][i];
                    lst[ft] = val[ f[ft]? f[ft]: lst[f[ft]] ];
                }
            }
        }
    }
    inline void prt(int i){   //打印同后缀结点号
        for( ; i; i= lst[i]) printf("%d\n",i);
    }
    void find(char*s){  //输出s中每个单词的结点号
        int p=0;
        int len= strlen(s);
        for_(i,0,len){      //遍历句中每个字母
            int id= s[i] -'a';
            while(p && !ch[p][id]) p= f[p];
            p= ch[p][id];
            if(val[p]) prt(p);
            else if(lst[p]) prt(lst[p]);
        }
    }
};
```